Amendments to the Drawings

The attached sheets of drawings includes changes to Figures 6 and 7. These sheets, which include Figures 6 and 7, replace the original drawing sheets including Figures 6 and 7.

Attachment: replacement sheet(s)

Remarks and Arguments

Claims 1-25 have been presented for examination. Claims 1, 11, 21 and 25 have been amended.

The drawings have been objected to by the official draftsman for ill-defined lines in Figures 6 and 7. New formal figures 6 and 7 have been submitted to correct these problems.

The specification has been objected to as incorporating an embedded hyperlink. In response, the specification has been amended by replacing the paragraph starting at page 8, line 19 to remove the hyperlink.

Claims 1, 11 and 25 have been provisionally rejected under the judicially-created doctrine of obviousness type double patenting over claims 1, 12 and 23 of co-pending application 09/965,218 and claims 1, 9 and 17 of co-pending application 09/960,122. The examiner comments that claims 1, 11 and 25 of this application and claims 1, 12 and 23 of co-pending application 09/965,218 and claims 1, 9 and 17 of co-pending application 09/960,122 differ only in the functions recited in the claims as being performed by the recited elements and that these functions are related and obvious in view of one another.

Since neither this application nor application serial number 09/960,122 have been allowed or issued, these rejections appear to be provisional. Accordingly, applicants will respond when the rejection is made final as set forth in MPEP §822.01.

Claims 1-25 were rejected under 35 U.S.C. §112, first paragraph as not being enabled by the disclosure. In particular, the examiner points to the interface layer recited in claims 1, 11, 21 and 25 as being mentioned in the specification at page 9, but not fully disclosed in a manner sufficient to enable the entire claim. The interface layers for volume configuration and for data services, including data caching, are illustrated in Figures 1A and 1B and described on page 5, line 1 to page 7, line 6. As indicated there, the storage volume interface layer and the additional data service layers are kernel drivers that are implemented in platform-specific code that can necessarily be accessed from, and control, only the host on which they are installed. Consequently, as described at page 9, line 16-24, a native interface layer is used to convert the platform-specific code into a platform independent language such as Java™. The platform

independent language is in turn converted into an object-oriented model by the management facade. The specification further describes these layers in detail and their method of operation. For example, the interaction of the federated beans, the management facade and the native interface to set up and control a data caching system is described in detail at page 19, line 32, to page 24, line 2, in connection with Figures 8-12.

Consequently, it is believed that the specification describes the data caching data service layer and the manner of using it in sufficient detail to enable one skilled in the art to make and use the invention. Accordingly, the rejection of claims 1-25 under 35 U.S.C. §112, first paragraph is hereby traversed.

Claims 1-25 have been rejected as being indefinite because the terms "the interface layer API" and "the management facade" recited in claims 1, 11, 21 and 25 lack antecedent basis. In response, claims 1, 11, 21 and 25 have been amended to clarify them and provide proper antecedent basis for all recited elements. Claim 1 is exemplary. The term "interface layer API" has been deleted. The term "the management facade" in line 14 has been changed to "the management facade software" which finds antecedent basis in line 9. Similar changes have been made to claims 11, 21 and 25. Accordingly, the rejection under 36 U.S.C. §112. second paragraph, is hereby traversed on the basis of the amendments made to claims 1, 11, 21 and 25.

Claims 1-7, 9-17, 19-23 and 25 have been rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 6,332,163 (Bowman-Amuah) in view of the admitted prior art in the specification at pages 1-2 (APA). The examiner comments that the Bowman-Amuah reference teaches caching web pages using an interface layer management facade software and a federated bean in a three tiered architecture. However, the examiner admits that Bowman-Amuah does not explicitly teach that the data is contained in a storage device with the interface layer located between the storage device and the network and controlling data flow thereto. Further, Bowman-Amuah does not explicitly disclose retrieving data from the storage device if the data is not in the host memory. However, the examiner comments that the APA discloses a storage device connected to a computer system by driver software that controls the flow of data

between a network and the storage device and also discloses retrieving data from the storage device if the data is not in the host memory.

The present invention relates to a three-tiered data caching system for use on a distributed computer system connected by a network. This system allows network management personnel to configure and control a data caching system running on a remotely-located host. In this system, storage devices are connected to the host by means of an interface that comprises low-level kernel routines written in platform-dependent code that run only on the host. The lowest tier of the caching system comprises management facade software running on the host that converts the platform-dependent interface written with the low-level kernel routines to platform-independent method calls. The middle tier is at least one data caching Java bean that also runs on the host. The data-caching bean communicates directly with a management GUI or CLI and is controlled by user commands generated by the GUI or CLI. The management GUI or CLI can be run anywhere in the network. Therefore, a manager can configure the data caching system from a single location and can cache individual volumes "on the fly" during ongoing data processing operations.

The Bowman-Amuah patent discloses a framework for building "netcentric" systems based on a client-server model operating over the Internet. The basic idea is that the framework provides various services that can be used in building a client server application. By this means, a client can access many different services such as web surfing, program downloads, etc. One of the framework services is a web server. As described, a web server interacts with a client to obtain web pages from various hosts on which the pages are located. In order to reduce the time that is necessary to obtain these pages, the web server temporarily stores or "caches" some of these web pages. If a subsequent request is made for a web page, and that page is in the web server cache, the page can be quickly sent to the client without having to reacquire it from the source host. In addition, another framework service provided by the Bowman-Amuah system is a terminal service that allows a user to connect to a remote server and interact with a program running on that server. Bowman-Amuah also describes using Java beans to implement these services. However, Bowman-Amuah discloses nothing

about controlling the web server cache service using the terminal service. Instead it merely provides the services and the design of the overall system is left to the user.

The claims have been amended to particularly point out the distinctions over Bowman-Amuah. Claim 1 is representative. It recites, in step (a), inserting an interface layer between driver software and a storage device, the interface layer exporting a platform dependent API comprising a plurality of API methods that can be called to control data passing between the driver software and the storage device. The examiner equates this interface to general component interfaces discussed in Bowman-Amuah. However, Bowman-Amuah does not disclose inserting interface layers between driver software and a storage device as recited in amended claim 1. Further the component interfaces disclosed in Bowman-Amuah are platform independent, not platform dependent as recited in claim 1.

Claim 1 also recites running, in the host computer system, management facade software that receives calls to platform-independent methods and generates at least one API method call to the interface layer in order to execute the platform-independent method calls. The examiner equates this to the use of the disclosed terminal services to emulate an interface required by a program running on a remote server. Bowman-Amuah does not disclose that the terminal services could be used to control components that, in turn, control data passing between a driver and a storage device as recited in claim 1.

Claim 1 further recites running, in the host computer system, a federated bean that generates platform-independent method calls to the management facade software to control the interface layer via the plurality of API methods. The examiner equates this to the mention of enterprise Java beans (EJB) in Bowman-Amuah. However, Bowman-Amuah merely states that EJB could be used to implement the services provided by the disclosed framework, it does not disclose how to use the Java beans in the manner recited in the claims and in particular, to instruct the interface layer to intercept requests for data from the storage device and, if the data is in the host memory, to retrieve the data from the host memory, and to retrieve the data from the storage device if the data is not in the host memory as recited in step (d) of claim 1.

The APA discloses a conventional disk caching system which is not remotely managed. Bowman-Amuah discloses a general framework for constructing web centric services. The combination of these two references cannot convert the conventional disk caching system of the APA into the remotely controlled disk caching system of the present claims because, while Bowman-Amuah provides a general framework for constructing systems of the type recited in the claims, it provides no guidance as to how these systems should be constructed. Thus, it cannot teach or suggest how to modify the disk caching system of the APA to provide the remote management capability recited in claim 1. Consequently, amended claim 1 patentably distinguishes over the cited combination of references.

Clams 2-7 and 9-10 are dependent, either directly or indirectly, on amended claim 1 and incorporate the limitations thereof. Consequently they distinguish over the cited combination of references in the same manner as amended claim 1.

Claim 11 has been amended in a manner similar to amended claim 1 and consequently patentably distinguishes over the cited combination in the same manner as amended claim 1. Claims 12-17 and 19-20 are dependent, either directly or indirectly, on amended claim 11 and incorporate the limitations thereof. Consequently they distinguish over the cited combination of references in the same manner as amended claim 11.

Claim 21 has also been amended in a manner similar to amended claim 1 and consequently patentably distinguishes over the cited combination in the same manner as amended claim 1. Claims 22-23 are dependent, either directly or indirectly, on amended claim 21 and incorporate the limitations thereof. Consequently they distinguish over the cited combination of references in the same manner as amended claim 21.

Claim 25 has also been amended in a manner similar to amended claim 1 and consequently patentably distinguishes over the cited combination in the same manner as amended claim 1.

Claims 8, 18 and 24 have been rejected under 35 U.S.C. §103(a) as obvious over Bowman-Amuah in view of the APA and further in view of U.S. Patent No. 6,298,478 (Nally.). The examiner comments that the Bowman-Amuah reference

combined with the APA teaches the limitations recited in claims 8, 18 and 24 except that the combination does not explicitly teach enabling a number of flusher threads with the number being determined by a Java bean. However, the examiner asserts that the Nally reference discloses using a Java bean to determine a number of flusher threads to use and that the combination of Nally with Bowman-Amuah and the APA would have been obvious in order to achieve good task management.

The Nally reference discloses a system for managing concurrent transactions between enterprise Java beans. The section to which the examiner points as disclosing that the number of flusher threads is determined by a Java bean actually discloses mechanisms by which the current transaction can be located. These mechanisms include lookup tables and pointers. However, no mention is made that any threads involved are flusher threads or that the number of flusher threads should be determined by a Java bean. Consequently, the addition of Nally to the combination of Bowman-Amuah and the APA cannot add any teaching concerning flusher threads as recited in claim 8, 18 and 24. Thus, claims 8, 18 and 24 patentably distinguish over the cited combination of Bowman-Amuah, the APA and Nally.

In light of the forgoing amendments and remarks, this application is now believed in condition for allowance and a notice of allowance is earnestly solicited. If the examiner has any further questions regarding this amendment, he is invited to call applicants' attorney at the number listed below. The examiner is hereby authorized to charge any fees or direct any payment under 37 C.F.R. §§1.17, 1.16 to Deposit Account number 02-3038.

Respectfully submitted

Paul E. Kudirka Date: 2/28/05

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656